Claims

We claim:

1. A buffered interface system for interfacing a continuous stream of serial TDM data to a network processor coupled to a parallel data bus, the interface system comprising:

an input port for receiving at least one stream of serial TDM data, each data stream comprising a continuous series of time-domain multiplexed time slots synchronized to a common frame pulse signal, each time slot corresponding to a respective virtual channel for carrying digital voice content;

a receive component coupled to the input port for buffering and assembling received TDM data so as to form bytes of parallel data; and

a parallel bus interface coupled to the receive component for transferring said bytes of parallel data from the interface system to a network processor via a connected parallel data bus.

2. A buffered interface system according to claim 1 wherein:

the receive component includes a serial-to-parallel converter for converting each time slot of the serial data stream into a corresponding byte of data,

the receive component further includes a receive memory for storing said data bytes;

the receive memory is organized so as to define at least two logical receive memory banks, and each of the receive memory banks is selectively configurable as either an active memory bank available for storing a series of said data bytes as provided by the serial-to-parallel converter, or as a non-active memory bank available for transferring previously stored data bytes to the parallel bus interface.

3. An interface system according to claim 2 wherein the receive component includes means for controlling the receive memory so as to store said incoming data bytes into an active one of the memory banks while concurrently transferring previously stored data bytes from the non-active memory banks to the parallel bus

interface, thereby processing real-time TDM data flow at the input port with minimum delay.

4. An interface system according to claim 3 wherein said receive memory comprises N+1 blocks of random access memory, arranged to configure a selected one at a time of said memory blocks as an active memory block while the remaining N memory blocks are configured as non-active for concurrently transferring previously stored data bytes to the parallel bus interface, N being a positive, non-zero integer.

5. An interface system according to claim 4 wherein each memory block includes at least 128 bytes for storing data corresponding to one frame of TDM data.

6. An interface system according to claim 4 wherein each memory block includes 128 16-bit words selectable as upper and lower bytes for storing data corresponding to one frame of TDM data from each of two streams.

7. An interface system according to claim 2 wherein said receive memory includes means for concatenating bytes across multiple non-active memory banks so as to form a wide word for parallel transfer to the parallel bus interface.

8. An interface system according to claim 2 wherein said receive memory includes 9 blocks of random access memory, arranged to configure a selected one at a time of said memory blocks as an active memory block while the remaining 8 memory blocks are configured as non-active, and wherein said receive memory includes means for unloading and concatenating read bytes across the eight non-active memory banks so as to form an eight-byte wide word for broadside transfer to the parallel bus interface, said wide word comprising 8 bytes of data corresponding to a selected one of said time slots.

9. An interface system according to claim 2 including an integer multiple M pages of random access memory, each page of memory comprising a corresponding N+1 blocks of random access memory as recited, each page arranged for buffering and assembling an additional stream of TDM data provided to the input port.

10. An interface system for interfacing a network processor coupled to a parallel data bus so as to generate a continuous stream of serial TDM data, the interface system comprising:

a parallel bus interface for connection to a parallel bus for receiving bytes of parallel data from a connected network processor;

a transmit component coupled to the parallel bus interface for buffering and arranging the received bytes of parallel data so as to form the stream of serial TDM data; and

a TDM output port for transmitting the stream of serial TDM data, the stream comprising a substantially continuous series of time-domain multiplexed time slots synchronized to a common frame pulse signal.

11. An interface system according to claim 10 wherein:

the transmit component includes a transmit memory for storing said received data bytes;

the transmit memory is organized so as to define at least two logical transmit memory banks, each transmit memory bank sized for storing a plurality of said data bytes for serialization into a frame of serial TDM data;

each of the transmit memory banks is selectively configurable as either an active memory bank, available for unloading stored data bytes, or as a non-active memory bank available for storing data bytes as they are received from the parallel bus interface; and

the transmit component includes a parallel-to-serial converter for converting each stored byte of data into a corresponding time slot to form the TDM data stream.

12. An interface system according to claim 11 wherein the transmit component includes logic for storing said received data bytes into the non-active memory banks while concurrently transferring previously stored data bytes from an active memory bank to the parallel-to-serial converter.

13. An interface system according to claim 12 wherein said transmit memory comprises N+1 blocks of random access memory, arranged to configure a selected N at a time of said memory blocks as non-active memory blocks while the remaining

memory block is configured as active for concurrently transferring previously stored data bytes to the parallel-to-serial converter, N being a positive, non-zero integer.

14. An interface system according to claim 11 wherein:

the parallel bus interface is coupled to the transmit memory for storing a multiple-byte data unit in a single, broadside write operation into multiple non-active memory banks, storing each data byte in a corresponding one of the non-active memory banks, so that a subsequent sequential read of a selected one of said memory banks produces a series of bytes corresponding to a frame of TDM data.

15. An interface system according to claim 14 and further comprising a cpu interface component including a control interface for control communication with a network processor; the control interface including at least one packet length register to allow network processor bus accesses of configurable length.

16. A system for interfacing a continuous stream of serial TDM data to a network processor coupled to a parallel data bus, the interface system comprising:

an input port for receiving at least one input stream of serial TDM data, each input data stream comprising a continuous series of time-domain multiplexed time slots synchronized to a common frame pulse signal, each time slot corresponding to a respective virtual channel for carrying digital voice content;

a receive component coupled to the input port an including a receive buffer memory for assembling received TDM input data so as to form first bytes of parallel data;

an output port for transmitting at least one output stream of serial TDM data;

a transmit component coupled to the output port and including a transmit buffer memory for disassembling the second bytes of parallel data so as to form the serial output TDM data; and

a parallel bus, coupled to the receive component for transferring said first bytes of parallel data to a connected network processor, and coupled to the transmit component for concurrently transferring said second bytes of parallel data from the connected network processor to the transmit component.

17. An interface system according to claim 16 and further comprising a cpu interface component including a control interface for control communication with a network processor.

18. An interface system according to claim 17 wherein the cpu interface component includes at least one control register.

19. An interface system according to claim 18 wherein said control registers include at least one a packet length register.

20. An interface system according to claim 17 wherein the cpu interface component includes a receive count (1660) register and a transmit count (1662) register to allow the network processor to monitor the receive and transmit component buffer memories, respectively.

21. An interface system according to claim 17 wherein the cpu interface module includes at least one status register for handshaking with the network processor.

22. An interface system according to claim 21 wherein the status registers include at least one of a Receive Count register, a Transmit Count register, and overflow status bits.

23. An interface system according to claim 16 wherein the receive component includes logic for notifying the network processor when data in the buffer memory is ready for unloading. 24. An interface system according to claim 16 wherein the receive component includes a standby buffer memory and logic for notifying the network processor when data in the buffer memory is ready for unloading; and further includes logic for storing data in the standby buffer memory while data in the buffer memory is unloading to the network processor.

25. A TDM bridge product comprising a circuit board configured for use in a host telecom system, the TDM bridge product comprising:

a first input connector mounted on the circuit board for receiving TDM input data according to a predetermined protocol, the TDM data comprising at least one stream consisting of a continuous series of time-domain multiplexed time slots, each

stream being synchronized to a common frame pulse signal, and each time slot corresponding to a respective virtual channel for carrying digital voice content;

a time slot switch means mounted on the circuit board and coupled to the first input connector for controllably selecting at least one time slot of the TDM input data and providing the selected time slot of serial data to a local bus;

a buffered interface component mounted on the circuit board and coupled to the local bus for receiving the selected TDM data and for forming wide data words, each wide data word including of a plurality of data bytes, and all of the data bytes within each wide data word corresponding to a selected time slot channel of the TDM input data;

a network processor bus implemented on the circuit board and coupled to the interface component to receive the wide data words;

a network processor mounted on the circuit board and coupled to the network processor bus for forming a series of data packets encapsulating the wide data words provided by the buffered interface component on the network processor bus; and

an output connector mounted on the circuit board for transmitting the data packets on to a packet switched network coupled to the output connector.

26. A TDM bridge product according to claim 25 wherein the wide data words each include no more than 8 bytes, corresponding to 8 frames of TDM data or 1 millisecond of sampled voice content for each active time slot.

27. A TDM bridge according to claim 26 including means for transmitting the formed data packets on to an IEEE 802.3 network.

28. A TDM bridge according to claim 26 including means for transmitting the formed data packets on to an ATM network.

29. A TDM bridge according to claim 26 including means for transmitting the formed data packets on to a SONET network.

30. A TDM bridge according to claim 26 wherein the wide data word consists of a number of bits equal to a number of data signal lines of the network processor bus for transfer to the network processor in a single read operation.

31. A method of bridging TDM data to a packet switched network with minimal delay so as to obviate echo cancellation, the method comprising:

receiving a stream of TDM data, the data stream comprising a continuous series of time-domain multiplexed time slots synchronized to a frame pulse signal, and each time slot corresponding to a respective virtual channel for carrying digital voice content;

converting bits of the TDM stream into a corresponding byte on each time slot boundary, thereby forming a continuous series of bytes corresponding to the TDM stream;

providing a plurality of N+1 memory banks, where N is a positive integer;

storing a first frame of the series of bytes into a first one of the memory banks;

storing each subsequent frame of the series of bytes into a next succeeding one of the memory banks, until N frames of data are stored in respective memory banks;

storing a next subsequent frame of the series of bytes in the N+1th memory bank; and

while storing said next subsequent frame of the series of bytes in the N+1th memory bank, concurrently unloading the first N frames of data from the first N memory banks into a processor;

responsive to a next frame pulse signal, rotating the memory banks; and then

repeating said steps of storing and unloading the series of data bytes in an ongoing fashion for continuous real-time operation; and

concurrently, in the processor, encapsulating the wide words of data so as to form a series of data packets bearing the TDM data; and

transmitting the series of data packets on to a packet switched network.

32. A method according to claim 31 wherein said unloading step includes:

reading a first byte of each of N memory banks in parallel, thereby reading N bytes of a first time slot;

concatenating all N bytes so as to form a wide word of the first time slot data;

writing the wide word of data to a processor; and

repeating said reading, concatenating and writing steps for each subsequent time slot in the stream.

33. A method according to claim 32 wherein each read step comprises reading a plurality of bytes of each of the N memory banks in parallel for simultaneous transfer to the processor.